Basic Arrow Network Concepts

For MC2 - 2002

By: Raphael M Düa
FAICD, FAPE, MACS(Snr), PCP, CP, Grad DISC
Owner & CEO
Micro Planning International Pty Ltd

P O Box 7177, 509 St Kilda Road
Melbourne Victoria 8004
Australia

Cell: +61 418 531111
raf@microplanning.com.au

THE UNIVERSITY OF MELBOURNE

Basic Arrow Network Concepts

A network is the pictorial representation of the Project Plan which shows the inter-relationships and inter-dependencies of the component tasks. It is sometimes known as a 'logic diagram' or 'arrow diagram', however the term 'critical path network' is more commonly used throughout industry and will be used for this paper

Critical Path Network Techniques

A network must logically express the sequence and pattern of work flow as well as the relationships and restraints implicit in the intended plan of operations. Since networks can be drawn for different levels of management, i.e. with greater or lesser detail, it is important to establish the nature and amount of detail to be included. Accurate network construction can often be achieved more readily as a team effort by key personnel having specific knowledge of the activities and processes involved. The leader of the team should be fully conversant with the rules and conventions applicable to network logic.

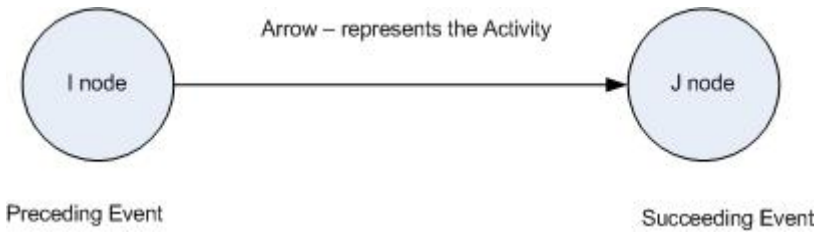
Critical Path Network Diagrams

A network can be represented by one of two techniques - arrow diagrams or precedence diagrams. However the general standard for the RPDP project is to use arrow diagrams; the basic element of which is the activity, which represents an amount of work to be performed. Each activity is given a duration, which defines the time required to complete the work as well as the resources require to carry out the work.

Networks can be subdivided into smaller units called sub-projects. Each sub-project must be logically complete. Sub-projects are the smallest unit that can be processed by Micro Planner X-Pert™ but they can be linked together to form larger networks.

Arrow Diagrams

The two elements of an arrow diagram are directional lines (or arrows) each representing one activity, and circles representing events. Events represent the points at which activities start and finish. Arrow diagrams are also known as I/j diagrams. See diagram below



The event which commences an activity is known as the Preceding Event (PE) and the event which terminates the activity is known as the Succeeding Event

In the figure below BUILD SECTION 1 and MAKE UP COMPONENT 2 finish at event E4, is a name given to an event and is known as the identifier by their preceding event and succeeding event identifiers. Therefore, in Figure.1 the first activity is identified as E1-E2, its description is PREPARE SITE and it has duration of four (4) time periods. Events may also be given descriptive data. For example, event E1 can be described as START OF NET1. However, as previously stated an event cannot have duration.

An event is achieved when its preceding activities are completed. For example, in Figure 1 event E4 (COMPONENT 2) and E2-E4 (BUILD SECTION 1) are complete.

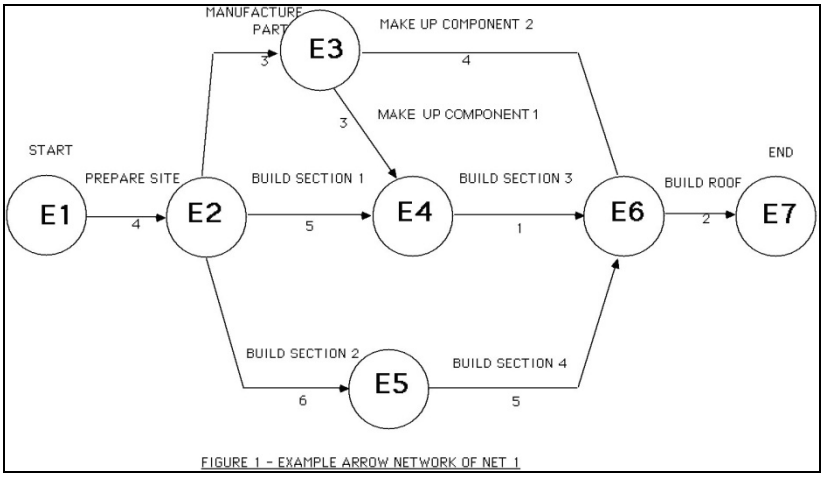


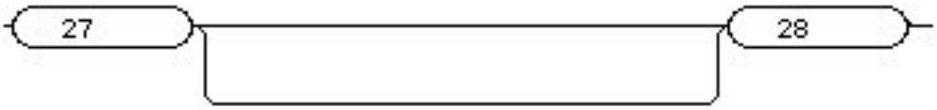
FIGURE 1 – EXAMPLE ARROW NETWORK OF NET 1

Accurate representation of the project by the network is very important, since the data that defines the network is the basis on which analysis is performed. The activities must be placed in logical work sequence, showing which may be worked at the same time. In Figure 1: MANUFACTURE PART A, BUILD SECTION 1 AND BUILD SECTION 2 can be worked at the same time, but cannot start until PREPARE SITE has finished.

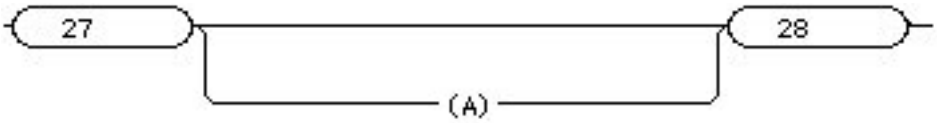
Activities may have properties that affect the way in which the network is processed. Such activities are described using the following activity types:

Parallel Activities and Activity Uniqueness

There is no reason why two activities cannot start and finish at the same events.



To maintain the activity uniqueness, *Micro Planner* alphabetically labels each parallel activity like this:



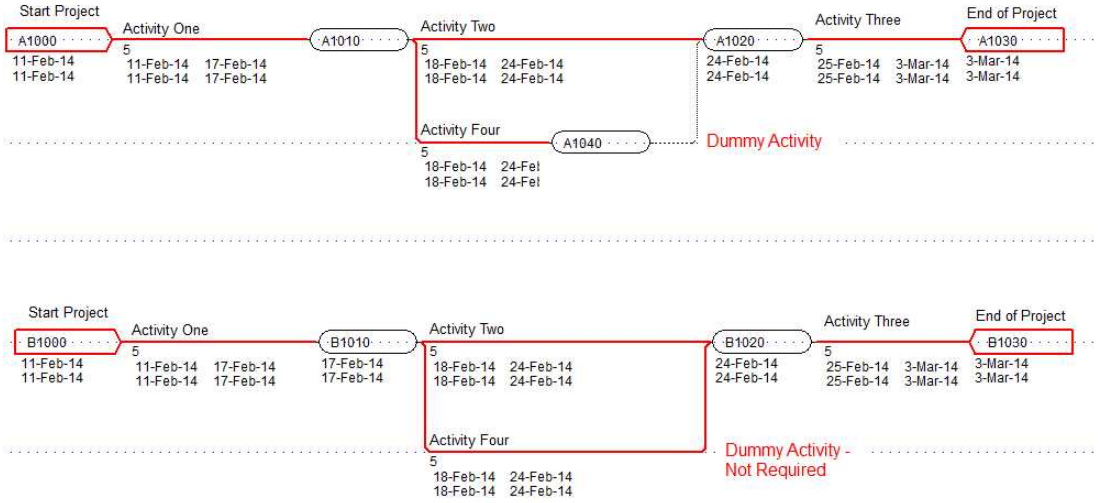
This **Uniqueness Identifier** is not always visible but is internally recorded.

Activity Types

DUMMY ACTIVITY

A dummy activity is one that represents a logical link between network paths rather than an actual task to be performed.

However, Micro Planner X-Pert does not need to specify a dummy to make parallel activties unnique, due to using “Uniqueness Identifier” as shown below



In arrow network diagrams, dummy activities are normally indicated by a broken line, as shown in Figure 2. below In this example, the SET UP ... activities and the BUILD ... activities can be worked in parallel. The network shows the normal constraints for BUILD STAGE 2; i.e. BUILD STAGE 1 and SET UP STAGE 2 must be complete before it can start. The BUILD STAGE 1 activity has a similar dependency shown by the dummy connecting E12 and E13. The network therefore shows that BUILD STAGE 1 cannot be started until SET UP STAGE 1 and PREPARE SITE are complete.

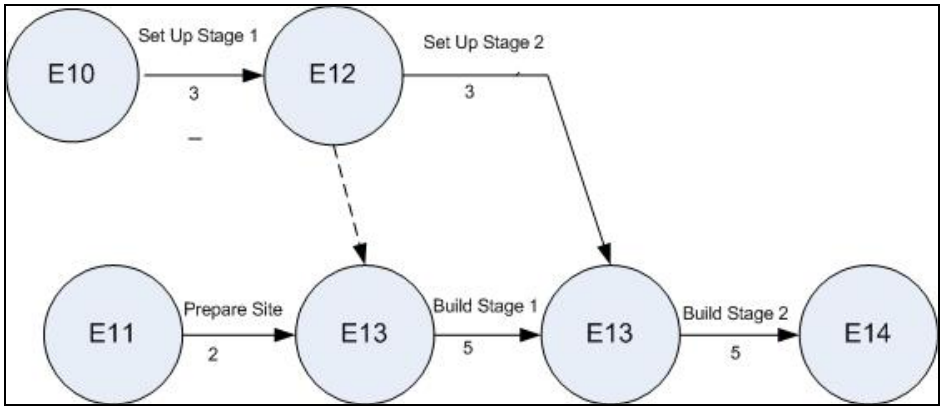


Figure 2 – Example of a Dummy

LADDER ACTIVITY

Ladder activities are a special group of activities that are used to represent progressive feed tasks; for example, in the manufacture of parts that are used to assemble components that are in turn used to make up finished products. Ladder activities are also known as a rung or ladder rung.

An example of a progressive feed task occurs in the manufacture of a number of identical components, each component having to go through several manufacturing processes. To represent these processes on the network in the normal way would require one activity for the manufacture of the components, another to assemble the unit, probably another for inspection, etc. The same sequence of activities would have to be repeated for each unit required. The resulting network could be extremely complex as shown in Figure 4.

Before the second activity in such a progressive feed process can start, the first activity must have been in progress for a given time to ensure a supply of components for the second activity. The time

that must elapse before the second activity starts is called lead time. similarly, there is a lag time after the completion of the first activity before the second activity can be completed. This situation is represented diagrammatically as follows:

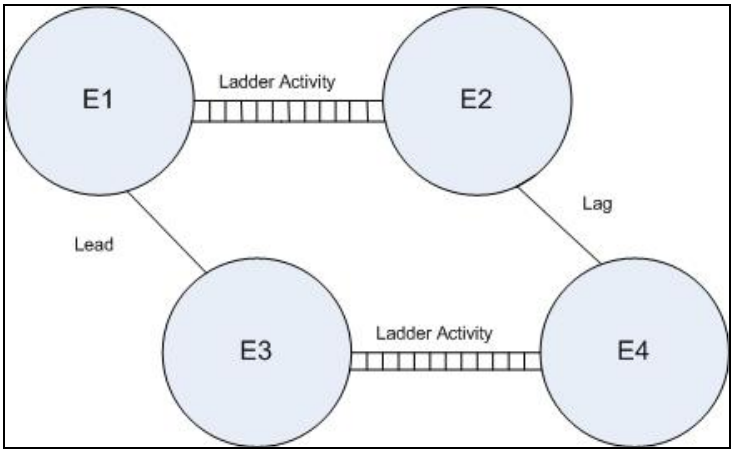


Figure 3: Example of a simple Ladder

Activity E1-E2 represents the first activity and E3-E4 the second. In the diagram, they are drawn horizontally and are known as the rungs of the ladder. Therefore the rung activities represent the progressive feed tasks.

Note: Micro Planner draws a ladder rung as two parallel lines shaded in order to highlight the fact that the activity is a ladder type. See Figure 3 above

Activity E1-E3 is a lead activity, and its duration is equal to the time that must elapse between the start time of the preceding and succeeding rungs. The duration of the lead determines the amount of the preceding rung that must have been worked before the succeeding rung can be started.

Activity E2-E4 is a lag activity, and its duration is equal to the lag time that must elapse between rungs. The duration of the lag determines the concluding amount of the succeeding rung that cannot be worked until the preceding rung has been completed.

In the case where multiple units of a product are being manufactured, then assembled and finally inspected the arrow diagram may well look like that illustrated in Figure 4 below

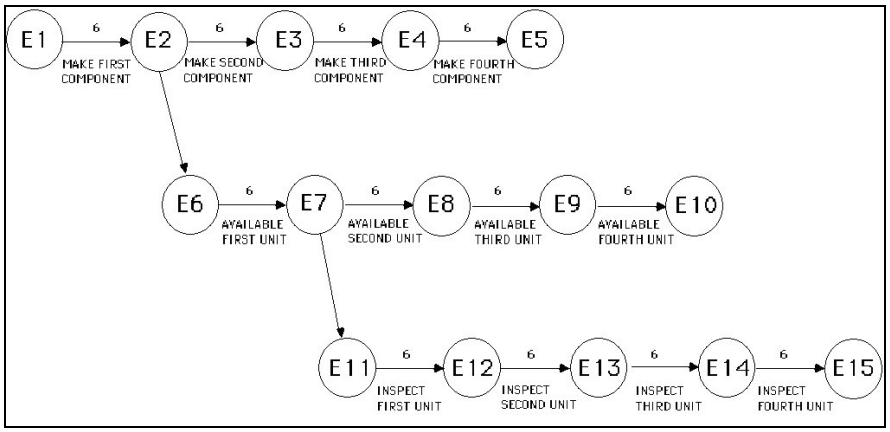


Figure 4 – Progressive Feed – Not in Ladder format

As soon as the first component E1 to E2 has been made , the succeeding dependent activity E6 to E7 to assemble the units now may start. The assembly of the first unit takes place and when completed

the succeeding activity E11 to E12 may commence. This process is known as progressive feed, but it is very difficult to defined just how many sub units the activity may be divided into. A far simpler view of this process is to use the Ladder format as shown in Figure 5 below

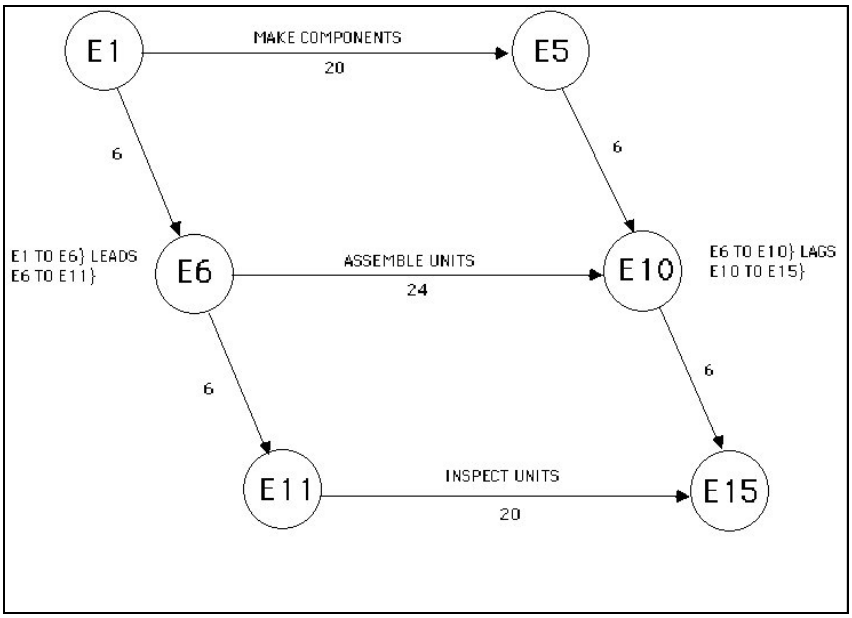


Figure 5 – Progressive Feed in Ladder Format

Figure 5 uses a ladder to represent the same progressive feed chain as Figure 4, but in a much simpler form. The ladder represents the following logic:

Acitivity E1 to E5, MAKE COMPONENTS, Activity E6 to E10 ASSEMBLE UNITS and Activity E11 to E12 INSPECT UNITS can all be worked in parallel, but ASSEMBLE UNITS cannot start until six time periods have been worked on MAKE COMPONENTS and cannot finish until six time periods after MAKE COMPONENTS has finished. INSPECT UNITS cannot start until six time periods have been work on ASSEMBLE UNITS and cannot finish until six time periods after ASSEMBLE UNITS has finished.

Ladder activities are normal activities given the “Ladder” activity type. However, leads and lags are special activities and must created by the user. The only data supplied by the user about a lead or a lag is its duration, it has no description field.

Therefore in the simple example given above in Figure 3, the durations for lead E1-E3 and lag E2-E4 would be associated with the Ladder activity E1-E2.

Complex ladder structures can be created. A ladder can extend to many levels of rungs and branches may occur in the structure. Therefore a rung may own several pairs of leads and lags, and may be the successor to many pairs of leads and lags. Simple examples of such structures are illustrated in Figures 6, 7 and 8 below.

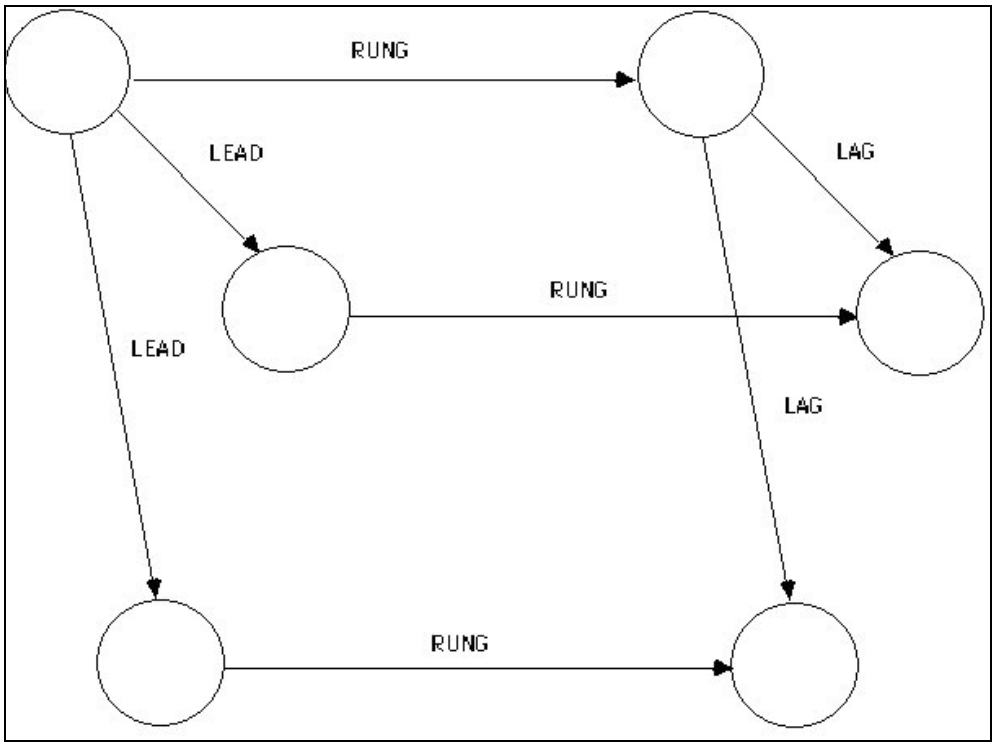


Figure 6 – Ladder with Multiple Rungs Lead and Lags originating from a rung

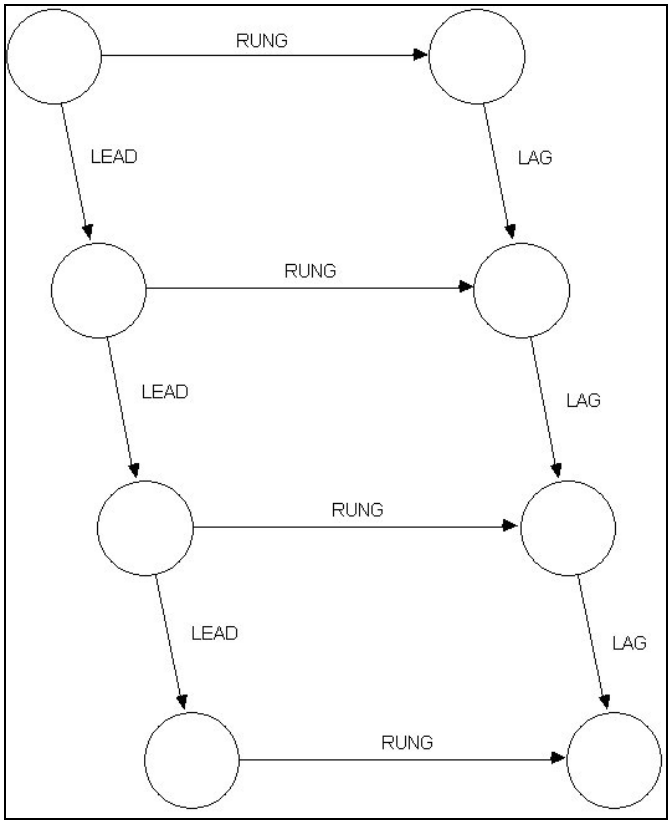


Figure 7 – A Four Rung Ladder

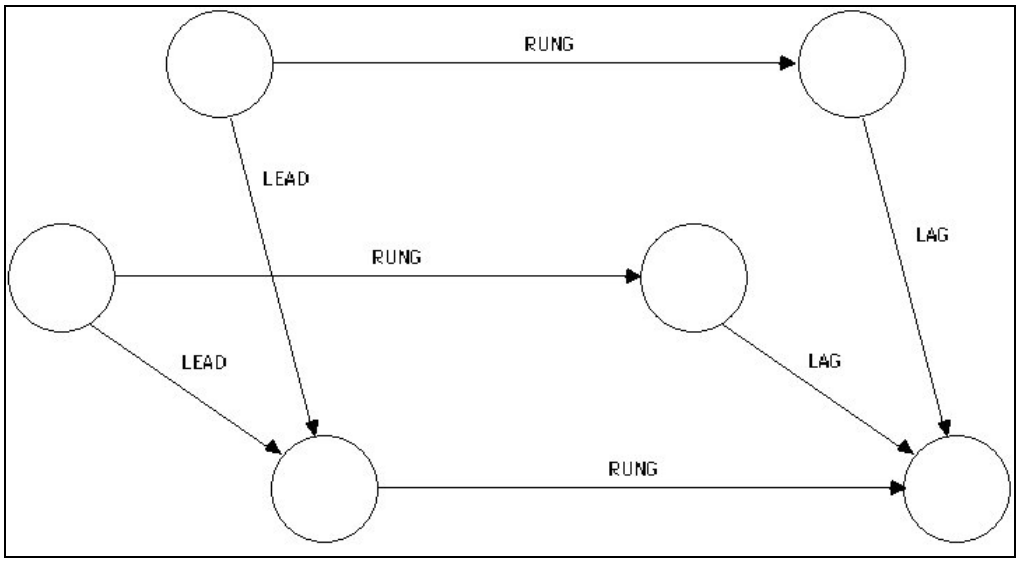


Figure 8 Rung with Multiple Rung Input

Note: The ladder technique was invented by International Computers and Tabulators Ltd in 1962, and has gained wide acceptance in network techniques.

Hammock

A hammock activity can be used to span a number of activities within a network. The duration of a hammock is not specified by the user; it is calculated by Time Analysis.

Hammock activities can be used in various ways, the two most usual applications being as follows: PRODUCING SUMMARY REPORTS If a project has for example, 12 major phases, each phase consisting of between 100 and 200 activities on the network, hammocks can be used to produce a high level report.

A hammock activity could span each phase of the network and a report selecting hammocks only would give the start and end time and duration of each phase.

SHOWING OVERHEADS Consider, the example, a project that requires specialist equipment to be hired for part of its duration. If a hammock activity is specified from the preceding event of the first activity that requires the equipment to the succeeding event of the last activity that requires it, the duration of the hammock will give the duration of hire.

In an arrow network, the preceding event of a hammock must have a non-hammock activity emanating from it. The succeeding event of a hammock must have a non-hammock activity entering it. See Figure 9.

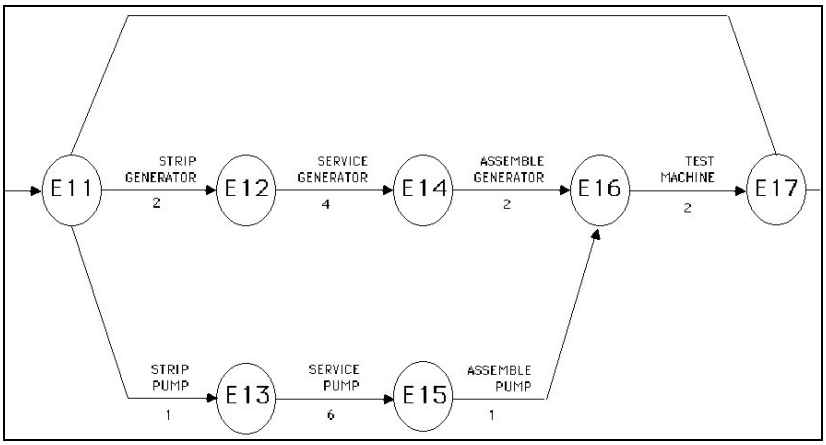
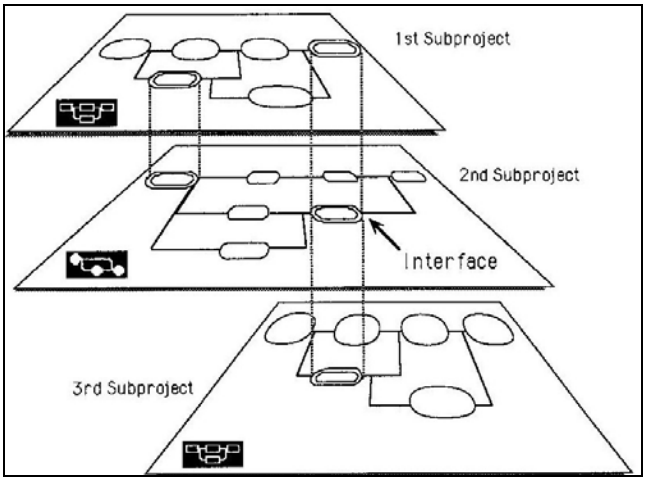


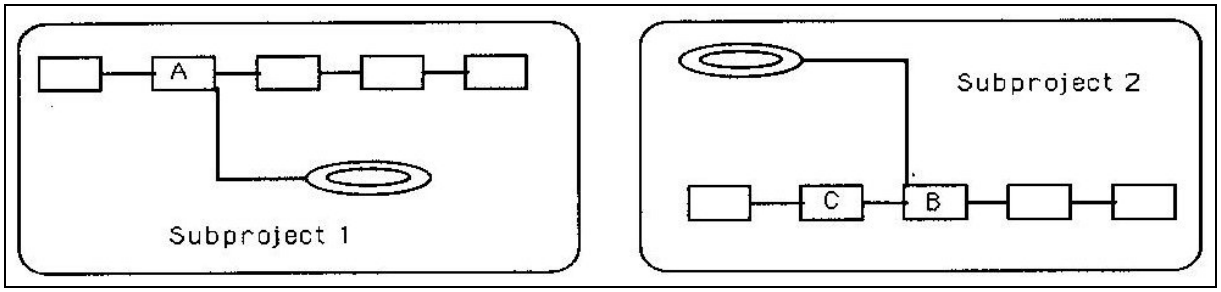
Figure 9 – Example of Hammock Activity E11 to E17
Duration is calculated by Micro Planner

Interfaces

Networks can be sub divided into smaller processing units called sub-networks (or subprojects). Interface events denote points in time at which tasks in different sub-networks (subprojects) are dependent on each other.



Logic will flow in **both** directions through the interface, in effect there is only one logical "point" although the interface is drawn in more than one subproject. If it is desired to control the direction of the logical "push", the following structure should be used:



This arrangement ensures that activity "B" is delayed by activty "A", but prevents any delays from "C" being passed back to Sunproject 1

Reverse Logic

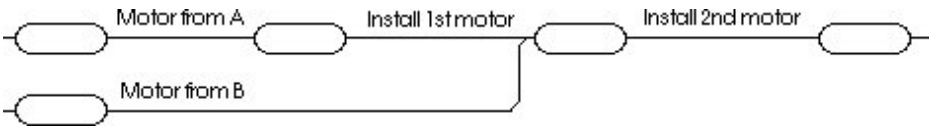
Having laid down strict rules of logic for building the network which states that no activity may commence until all preceding dependent activities have been completed, there is an exception. The Reverse Logic Node, as its name suggests, reverses the normal logic from: An operation can only start when the last of its predecessors is complete. This is also known as the Short Path or OR node logic. Which says that:-

An activity can start as soon as the first of its predecessors is complete.

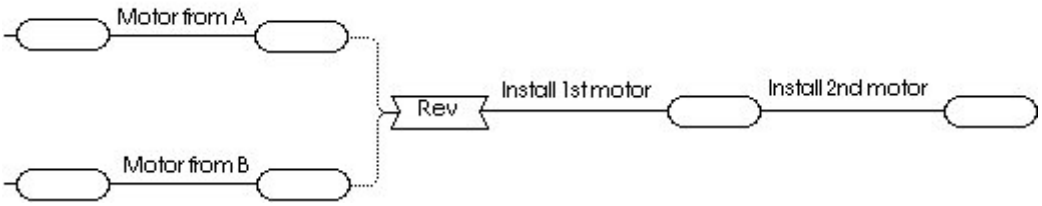
Consider the following

A project involves the delivery of two motors, which, for reasons of space, must be installed one after the other. These motors are interchangeable but obtained from different suppliers. You do not know, when constructing the network, which motor will arrive first.

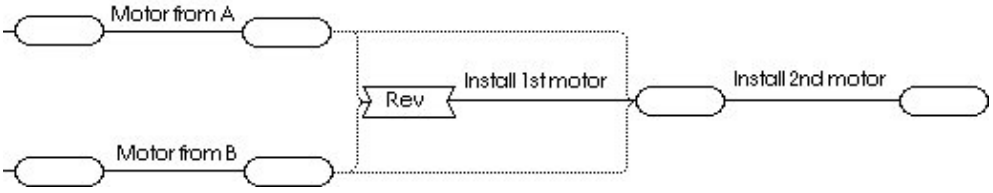
This example assumes that the motor from A will be the first one delivered -- but what if the motor from B arrives first?



By introducing two Dummies and a Reverse Logic node, the first motor to be delivered - irrespective of the supplier - is the first one to be installed...



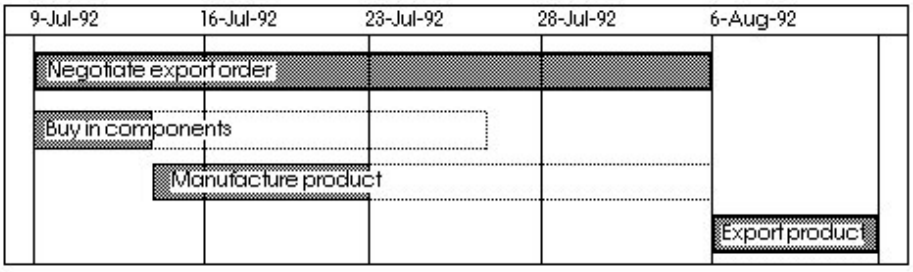
Finally, two more Dummies are used to connect the delivery of both motors to the 2nd installation -- to ensure that this activity does not start until the second motor has actually been delivered!



Wait Activities

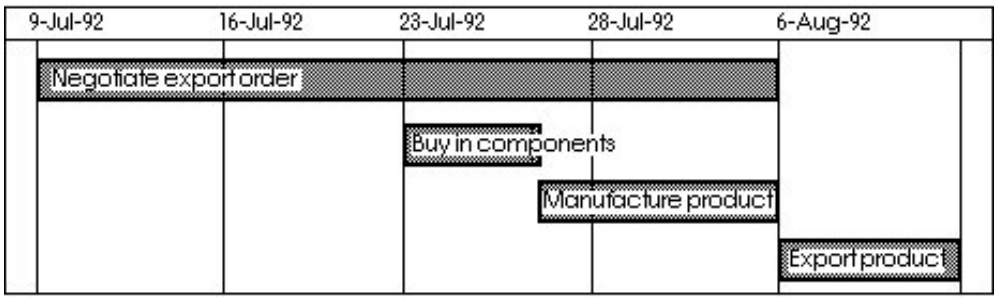
Micro Planner when scheduling through Time Analysis schedules an activity to start as soon as its predecessors have finished.

However, there may be times when you switch the emphasis so that an operation finishes *just in time* for its successors to begin. In the example below:-



A project that involves the manufacture and export of a product, by leaving Time Analysis to its own devices, this product is ready long before it can be exported - resulting in costly storage charges and cash flow implications.

By specifying 'Buy in components' and 'Manufacture product' as a *Wait* activity, the user can re-schedule these operations to finish just in time for export like this.



Critical Path Calculations

In an Arrow network, the longest path through to the end activty in terms of time is known as the 'Critical Path'. This path is obtained by the addition of all activity times which have been estimated along each and every path in the network. The calculations by which these paths are computed are described in the succeeding paragraphs.

Event Calculations

Event Early Schedule
Time (EES) is the earliest that succeeding activities may start.

Early Time
(ET) is the earliest that succeeding activities may start, based on ALL previous Management Impositions (M.I.).
(M.I.'s are imposed schedule dates).
ET = Max (EES, ESS + d)

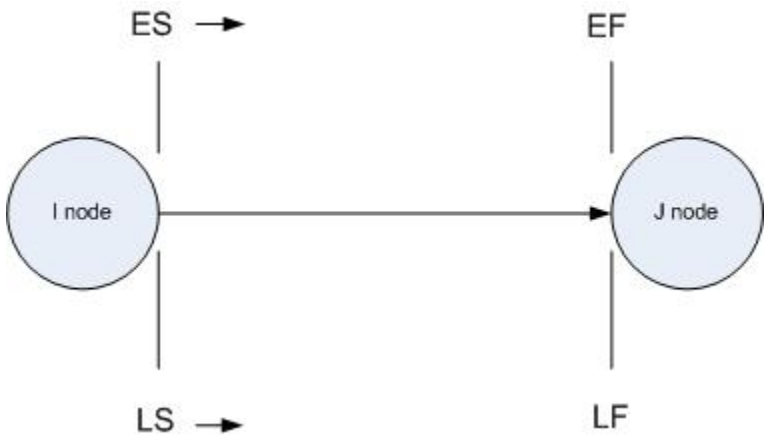
Late Event Schedule
Time (LES) is the latest that the succeeding activities may start.

Late Time
LT is the latest that the most critical succeeding activities may start.
LT = Min (LRS, LES =

Slack
(SK) is the amount of time that an event occurrence may be delayed from its earliest time.
SK = LT,

Activity Calculations

Earliest Start Schedule time (ESS) is the earliest that an activity may start.



Earliest Start
(ES) is the earliest that an activity may start, based on all previous MI.
ES = Max (EES, PPE)

Latest Finish Schedule
Time (LFS) is the latest that an activity may finish.

Latest Finish
(LF) is the latest that an activity may finish, based on all Subsequent MI.
LF = Min (LFS, SEL)

Latest Start
(LS) is the latest that an activity may start.
LS = LF - A

or LS = PEL
(for activities designated Ladder)

Earliest Finish
(EF) is the earliest an activity may finish.
EF = ES + A
or EF = SEE
(for activities designated Ladder)

Total Float
(TF) is that amount of time available from earliest start, in addition to the estimated activity time, that may elapse without violating any management impositions.

TF = LF - (ES = A)

This includes the waiting time due to uneven work load on a ladder.

Early Free Float
(EEF) is that amount of time available from the earliest start, in addition to the estimated activity time, that may elapse without affecting the earliest times of any subsequent activities or events.

EFF = Min (LF, SEE) - (ES + A)

Late Free Float
(LFF) is that amount of time available from the preceding event latest, in addition to the estimated activity time, that may elapse without affecting or violating any management impositions.

LFF = LF - A - Max (PEL, ES)

Independent Float
(IF) is that amount of time available from the preceding event latest, in addition to the estimated activity time, that may elapse without affecting the earliest times of subsequent activities and events.

3. Example Calculation
To show how a network is analyzed the example shown in Figure 10 will be calculated.

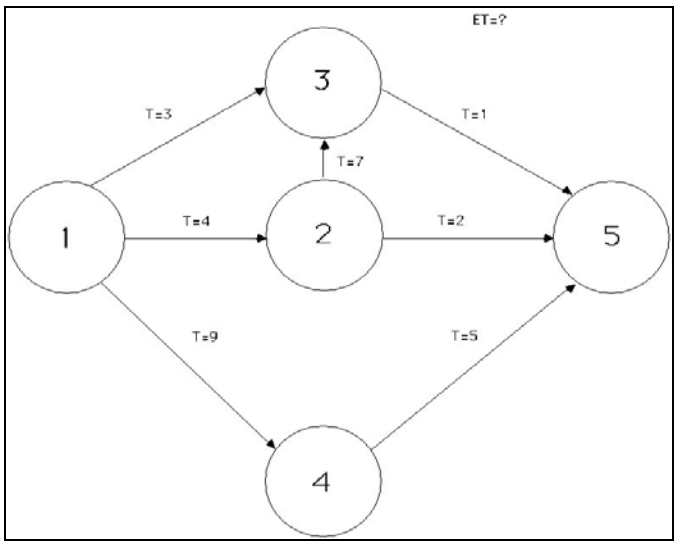


Figure 10. Example Network for Calculations

To determine the earliest finish date of event 5 (the end of the network), all the estimated activity times must be summated along each path. There are four paths leading to 5:

1	to	3	to	5			3	+	1	=	4	
1	to	2	to	3	to	5	4	+	7	+	1	= 12
1	to	2	to	5	and the sum or t		4	+	2	+	6	
					for each part is:							
1	to	4	to	5			9	+	5	+	14	

ET then, is equal to the largest sum, namely 14.
Since all paths leading to event 5 must be completed before event 5 is achieved, then the longest elapsed time determines when event 5 will be attained; other paths of activities will be accomplished before and concurrently with the longer path.

In the summation of the activity times along the paths indicated, it should be noted that path 1 - 3 - 5 is not a real possibility; at least it is not possible as stated. Event 3 cannot be attained simply by completing activity 1 - 3, but requires that activities 1 - 2 and 2 - 3 also be completed. Therefore, 11 units of time must be expended, rather than 3 to accomplish event 3.

Since this is the case, it is impossible to reach event 5 in 4 units of time as shown above. This illustrates the fact that in computing ET for all the events in the network, the ET's for the earlier events must be calculated prior to the ET's for the later events.

This process of calculating the ET's is known as the Forward Pass

Accordingly if the ET's were calculated for all events in the network shown above in Figure 10, the results would be:

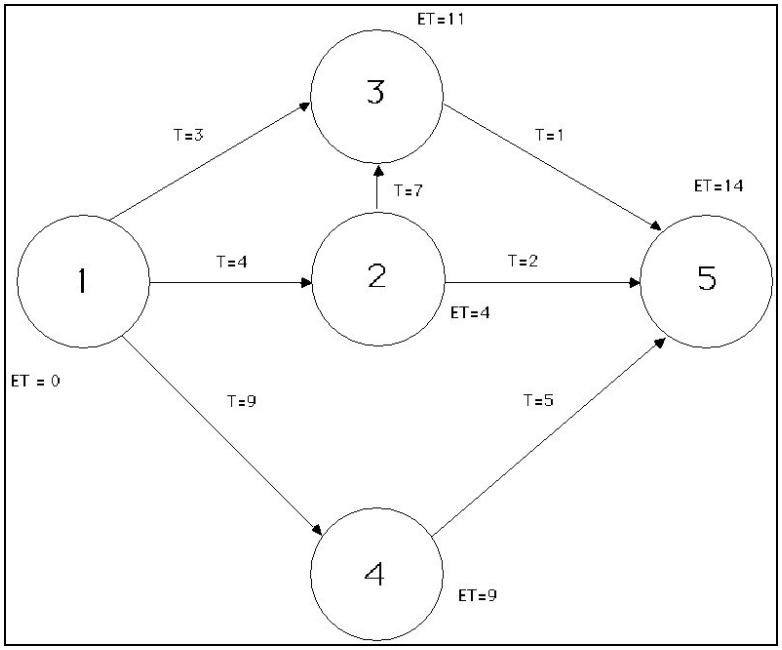


Figure 11 – Forward Pass Calculations for ET

It has been previously stated that the longest path in terms of time is known as the Critical Path. This is the path with the greatest time constraint on the end event, so it will determine the date upon which the end objective can be obtained. If any activity on this path requires time in excess of its original estimates, the entire program will be delayed by this amount. In the network shown in Figure 11 , the Critical Path is form 1 to 4 to 5.

On the remaining paths there is 'time to spare', or float at least relative to the end objective. These paths are called float paths. In order to compute the amount of float for each path, it is necessary to determine the latest time at which the path can be completed without itself becoming critical. If the expected time (ET) for each event is subtracted from the latest time (LT) for each event, then the amount of float for each event is determined.

Suppose it was decided that the expected time for event 5 was also the latest time that project conditions would allow it to be obtained. The determination of what is the latest allowable time is an arbitrary decision made independently of the network by Project Management. Suppose that $LT_i = LT_j - dur$ then

(1) By subtracting the activity duration (t) from the LT of its succeeding event, the LT of the preceding event is obtained. In the network shown in Figure 6.4, the activity duration of Activity 4 to 5 (t = 5) is subtracted from the latest allowable time of event 5 (LT = 14) to arrive at the latest allowable time for event 4 (LT = 9)

(2) Where there is more than one path leading back to an event, more than one LT is obtained. The smallest value is the correct one. This is true at event 2 in the network shown in Figure 6.4, where there is a path from event 5 back to event 2, and another path from event 5 back through event 3 to event 2.

In the sample network, if rules 1 and 2 are applied, LT for each event would have the following values as shown in Figure 12 below

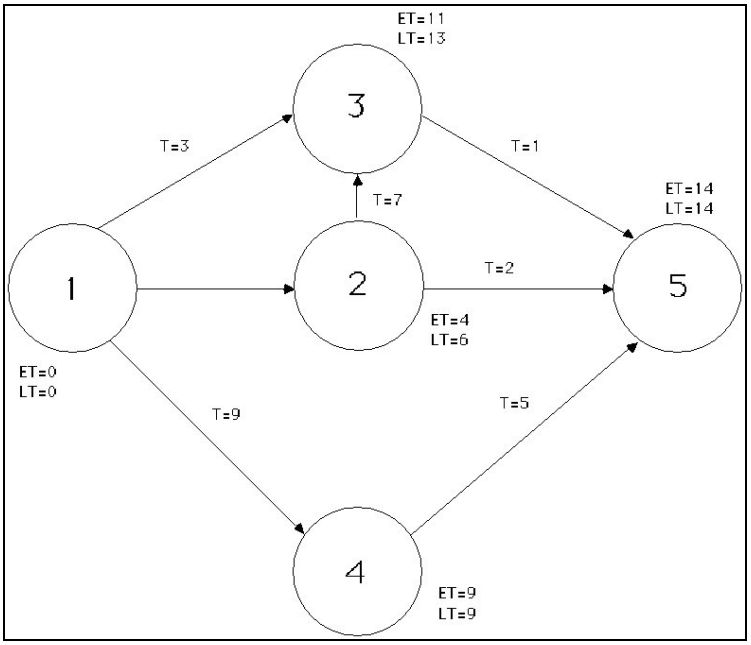


Figure 12 Backward Pass Calculations for LT's

Having determined LT for each event, it is now possible to determine slack for each event. Slack is equal to the difference between LT and ET i.e., slack = LT - ET for any event. Slack values for the network shown in Figure 12 are as follows:

EVENT	LT	ET	SLACK (LT-ET)
1	0	0	0
2	6	4	2
3	13	11	2
4	9	9	0
5	14	14	0

Note that event 1, 4 and 5 have no slack; this equates to stating that they are on the Critical Path. Therefore, if they are not accomplished by the expected time, not only they, but the entire project will become delayed.